演化計算 Hw2

> — 0416235 劉昱劭
>
> 點擊連結觀看網頁版：https://hackmd.io/s/r1F1o3dqE

1.

> Discuss why steady-state GAs and ES form two extremes regarding the population size and the number of offspring created.

- 我們知道：
  - mutation 是對單一 individual 的 operator
  - crossover 需要至少兩個 indivdual
- 最大的差異是產生子代的運算中，crossover 的有無
  - GA (幾乎) 一定有 crossover。
  - ES 可以不要 crossover，主要依靠 mutation。
- ES 產生子代的方式以 mutation 為主，mutation 只要有一個 individual 就可以 work。
  - 因此 ES 的 population size 可以很小，例如 (1+1)-ES 的 population size =1 。
  - 子代的數量通常跟 population size 相同 (一樣小)，因為每個 parent 會 mutate 出一個 offspring。
- GA 產生子代依靠 crossover 和 mutation。
  - 因 crossover 需要多一點 individual 來提供多樣性，以探索 search space 的其他地方，所以 GA 的 population size 會比 ES 大得多。
  - offspring size 大約是 population size 的大小，會比 ES 大得多。

> ES 的行為比較像 DFS
> GA 的行為比較像 BFS

2.

- 不管 one-point crossover 還是 uniform crossover 都不會改變其 0, 1 出現的 frequency。
- 這種 crossover 可以視為變換 allele 的 combination，只是交換到不同 individual 上，並不會增減 0, 1 的數量。

minimize $f(x) = \sum_{i=1}^{n} x_i^2, \quad x_i \in R, \quad i = 1, 2, \ldots, n$

3. `(1+1)-ES` , `(1,1)-ES` with fixed σ

`(1+1)-ES`	0.01	0.1	1.0	`(1,1)-ES`	0.01	0.1	1.0 `
Run #1	925	259701	∞	**Run #1**	∞	∞	∞
Run #2	814	263263	∞	**Run #2**	∞	∞	∞
Run #3	856	27812	∞	**Run #3**	∞	∞	∞
Run #4	891	726314	∞	**Run #4**	∞	∞	∞
Run #5	835	5442	∞	**Run #5**	∞	∞	∞
Run #6	881	899414	∞	**Run #6**	∞	∞	∞
Run #7	788	27931	∞	**Run #7**	∞	∞	∞
Run #8	795	141405	∞	**Run #8**	∞	∞	∞
Run #9	898	216004	∞	**Run #9**	∞	∞	∞
Run #10	862	139607	∞	**Run #10**	∞	∞	∞

4. cf. `(1+1)-ES` , `(1,1)-ES`

4-1. `(1+1)-ES`

- 每次 survivor selection 的時候，就能從 parent 和 offspring 之間挑一個 fitness 好的，這件事能讓 population 往 optimum 靠近。
    - 如果 offspring 比 parent 還糟，那表示這個 step 不好，留下 parent，下一代再 mutate 一個新的 step。
    - 但對這個問題來說，step size 給到 1.0 已經太大了，遠大於我們所設的門檻 0.005， `(1+1)-ES` 同樣收斂不了。
        - 當 step size=1.0，表示每一維的 step 期望值會是 1，很有可能他會在最佳解附近繞圈，但是很難走進我們設定的 threshold 0.005，這麼大的 step size 很容易走超過。
- 由結果可知，此題的 step size 在 0.01 這種量級是最快收斂的，0.1 次之，到 1.0 幾乎已經無法收斂。

4-2. `(1,1)-ES`

- 可以說是 random search，沒有辦法預測下一代會往哪裡 mutate，也沒辦法保證它會往 optimum 收斂，所以可以看到結果全部收斂不了。

5. `(1+1)-ES`, `(1,1)-ES` with self-adapted σ_i

- 參數設定：
  - $\tau = 0.1 * \frac{1}{\sqrt{2n}} \approx 0.02$
  - $\tau' = \frac{1}{\sqrt{2\sqrt{n}}} \approx 0.3976$
  - $\epsilon_0 = 10^{-5}$

`(1+1)-ES`	0.01	0.1	1.0	`(1,1)-ES`	0.01	0.1	1.0
Run #1	2120	2294	∞	**Run #1**	∞	∞	∞
Run #2	2173	2282	∞	**Run #2**	∞	∞	∞
Run #3	1845	2858	∞	**Run #3**	∞	∞	∞
Run #4	3122	2133	∞	**Run #4**	∞	∞	∞
Run #5	4128	2127	∞	**Run #5**	∞	∞	∞
Run #6	1920	2126	∞	**Run #6**	∞	∞	∞
Run #7	2161	2129	∞	**Run #7**	∞	∞	∞
Run #8	3130	2133	∞	**Run #8**	∞	∞	∞
Run #9	1704	2124	∞	**Run #9**	∞	∞	∞
Run #10	2133	3088	∞	**Run #10**	∞	∞	∞

6. cf. 3, 5

- 第五題的方式會有全體 (10 維共同) τ，跟自己 (每一維) τ' 兩個 learning rate，有可能比較靠近極值的維度，被其他維給拖走，導致整個 individual 無法靠近 optimum。
- 因此我把全體的 learning rate τ 調低，以減少這種現象。

6-1. `(1+1)-ES`

> 註：代數 = generation count

- 相較於第三題，可以看到 0.01 所需的代數增加，0.1 則減少。
  - 由第三題的結論推測，這組參數容易把平均的 step size 調至 [0.01, 0.1] 之間。
- 但 step size 初始值是 1 的好像還是 step 過大，無法收斂到半徑 0.005 之內。
- 當 ϵ_0 設過小 ($< 10^{-5}$) 的時候，會增加收斂所需要的代數。

6-2. `(1,1)-ES`

- 一樣是 random search，無法保證其往極值靠近。

7. 1/5-rule

- 參數設定：
    - G = 35
    - a = 0.89

`(1+1)-ES`	0.01	0.1	1.0	`(1,1)-ES`	0.01	0.1	1.0
Run #1	527	726	1057	**Run #1**	∞	∞	∞
Run #2	579	623	1123	**Run #2**	∞	∞	∞
Run #3	516	502	1261	**Run #3**	∞	∞	∞
Run #4	657	589	1167	**Run #4**	∞	∞	∞
Run #5	588	627	1114	**Run #5**	∞	∞	∞
Run #6	666	531	1180	**Run #6**	∞	∞	∞
Run #7	629	602	1017	**Run #7**	∞	∞	∞
Run #8	626	489	1081	**Run #8**	∞	∞	∞
Run #9	534	546	1158	**Run #9**	∞	∞	∞
Run #10	543	468	1130	**Run #10**	∞	∞	∞

8. cf. 3, 5, 7

- 這種 self-adpated mutation 比起第五題更能迅速修正。
- 第五題的 n-step mutation 可能會導致不同維互相影響，可能無法有效的靠近 optimum，而 1/5 rule 在一直無法靠近 optimum 的時候 (即 Ps 小)，以減少 step size，避免在 optimum 附近繞圈而進不去門檻內的問題。

8-1. `(1,1)-ES`

- 因 1-5 rule 可動態伸縮 step size，即使是原本 step size=1 過大，也會慢慢的把 step 調小以致於收斂。

- 而這種比較彈性的 step size 也讓 0.01, 0.1 的 case 更快的收斂到極值。

    - 能用比較大的 step size explore 大範圍的 search space
    - 走到極值附近之後，以小的 step 逼近極值以走進門檻的限制。

8-2. `(1,1)-ES`

- 一樣是 random search，無法保證其往極值靠近。